Unsupervised Statistical Learning: Clustering with k-means

Siva Balakrishnan
Data Mining: 36-462/36-662

March 21st, 2019

Recap: Principal Components Analysis (PCA)

- In unsupervised learning, we are just given a (big) data matrix $X \in \mathbb{R}^{n \times p}$.
- A basic question is: can we (meaningfully) reduce the dimension of the data either so we can visualize it, cluster it, or even do better supervised learning with it.
- PCA answers this questions by finding "interesting directions" and projecting the data on to those directions.
- It is the most widely used exploratory data analysis tool. It is extremely useful!

Recap: What are principal components?

- The linear algebraic answers: $X \in \mathbb{R}^{n \times p}$ _ mean 0.
  1. They are just eigenvectors of the covariance matrix $\underline{\widehat{\Sigma}} = X^T X / n$. → symmetric.

     $\hat{\Sigma} = VDV^T$. ~ PCs just cols. of V.

  2. Equivalently, they are the right singular vectors of the matrix $\underline{X}$.

     $X = U\tilde{D}V^T$ → just cols of $\bar{V}$.

Recap: What are principal components?

- The statistics/data-based answers:
  1. They are the directions of maximum variance. For instance, the first principal component:

$$v_1 = \arg\max_{\|v\|_2=1} \underbrace{v^T \hat{\Sigma} v}.$$

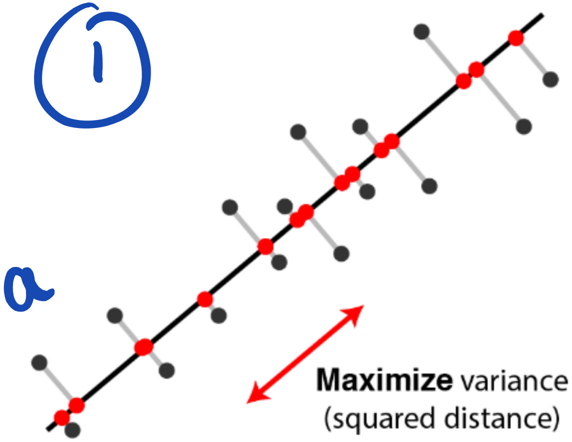
$\hookrightarrow$ measuring variance along v.

  2. They are subspaces which are closest on average to the data.

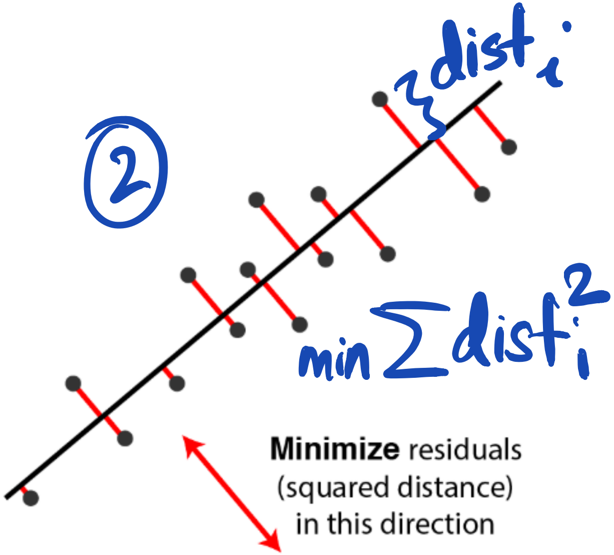
b

a

$\underline{(b^T a)}\, a$



Maximize variance (squared distance) of red dots in this direction



$\min \sum \text{dist}_i^2$

Minimize residuals (squared distance) in this direction

Amount/Proportion of Variance Explained

Eigenvalues are variance explained.

- Suppose we write:

$$\widehat{\Sigma} = VDV^T$$

then the:

1. Total variance in the data is given by:

$$T = \text{sum. of diagonals of } D.$$

2. Variance explained by i-th principal component is given by:

$$d_{ii}$$

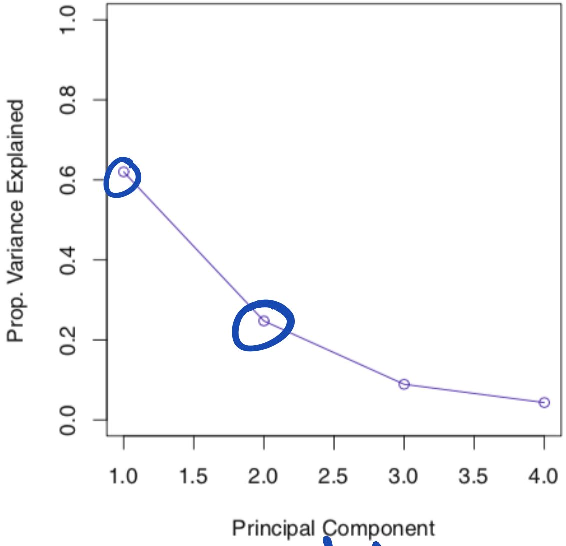
3. Cumulative proportion of variance explained:

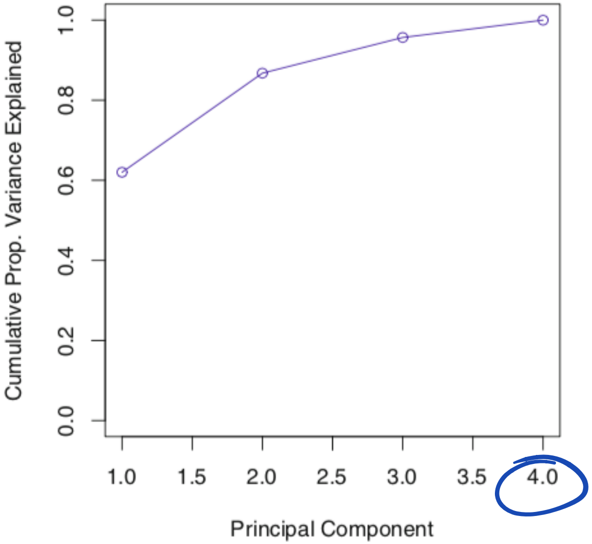
$$= \frac{\sum_{i=1}^{k} d_{ii}}{\sum_{i=1}^{p} d_{ii}} \Big\} \text{total variance.}$$

Recap: Amount/Proportion of Variance Explained

- Leads to two visualizations of the value of added principal components:

cum prop. of variance exp





Scree plot.

Recap: Dimension Reduction/Visualization

- Suppose we want to visualize our data (or reduce its dimensionality) in k dimensions.
- We simply compute the projection of our data onto the k PCs and plot it: $Xv_1 \in \mathbb{R}^n$, $Xv_2 \in \mathbb{R}^n$.

  plot rows of (Xv_1, Xv_2).

  $X = U\tilde{D}V^T$

  $Xv_1 = u_1 \times \tilde{d}_{11}$, $Xv_2 = u_2 d_{22}$

  These are called PC scores.
- If for some reason we wanted to plot it in the original space (i.e. plot the reconstruction of the data in the subspace spanned by the PCs):

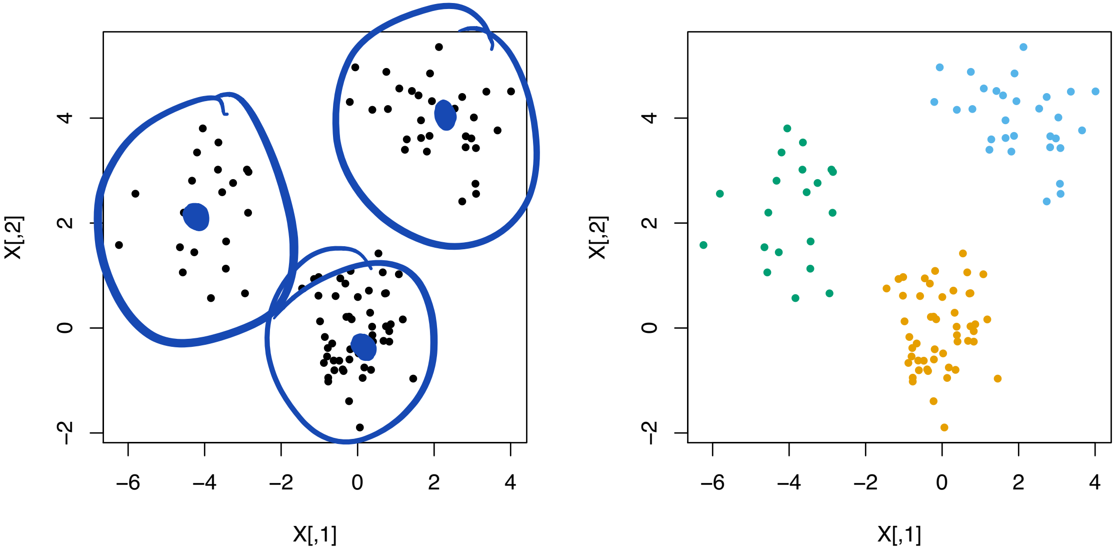
  $\tilde{X} = Xv_1v_1^T \in \mathbb{R}^{n \times d}$.

  $\hookrightarrow$ rank 1 matrix.

Just to get a (short) break from linear algebra we'll talk a bit about clustering today and then return to more dimension reduction (and lots more linear algebra)!

What is clustering? And why?

Clustering: task of dividing up data into groups (clusters), so that points in any one group are more “similar” to each other than to points outside the group



Why cluster?

Why cluster? Two main uses

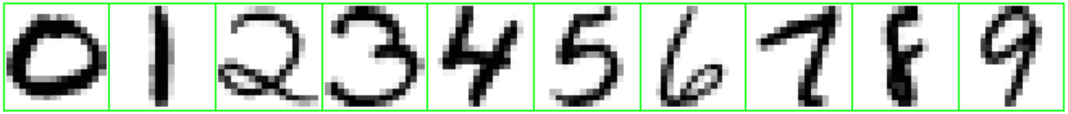
- Summary and data compression: deriving a reduced representation of the full data set.
- Discovery: looking for new insights into the structure of the data. E.g., finding groups of students that commit similar mistakes, groups of clients with similar behaviors, groups of assets with high dependence, or groups of users with similar behaviors/likes/clicks.

Other uses, e.g.,

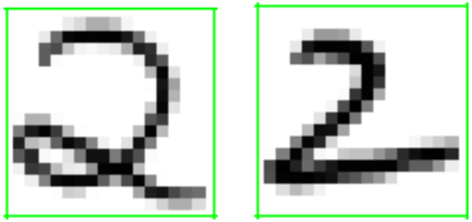
- Helping with prediction, i.e., in classification or regression
- Active learning, i.e. reducing the number of labeled examples we need to do supervised learning

Don't confuse clustering and classification!

In classification, we have data for which the groups are known, and we try to learn what differentiates these groups (i.e., classification function) to properly classify future data



In clustering, we look at data for which groups are unknown and undefined, and try to learn the groups themselves, as well as what differentiates them



Some terminology and notation

Given observations $X_1, \ldots X_n$, and dissimilarites $d(X_i, X_j)$. (E.g., think of $X_i \in \mathbb{R}^p$ and $d(X_i, X_j) = \|X_i - X_j\|_2^2$)

$\in X_i \in \mathbb{R}^p$.

n x n matrix.

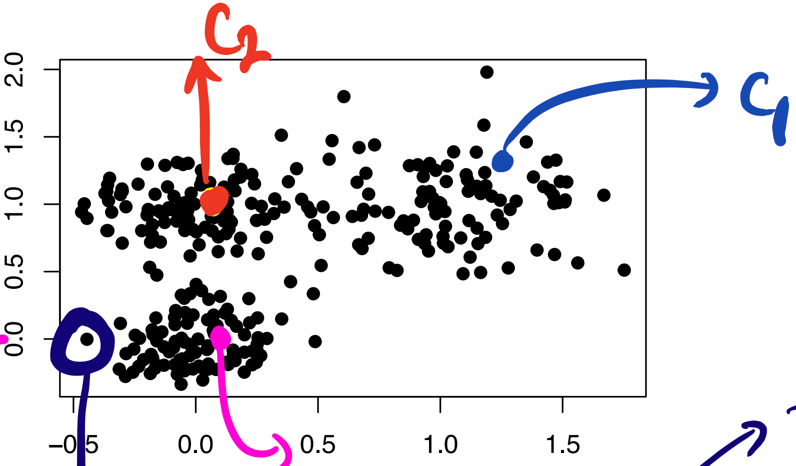
sq. Euclidean dist.

Let K be the number of clusters (fixed). A clustering of points $X_1, \ldots X_n$ is a function C that assigns each observation X_i to a group $k \in \{1, \ldots K\}$

$C(X_i) \in \{1, \ldots K\}$.

clustering that min

$$\sum_{i: C(i)=k} \|X_i - C_k\|_2^2$$



where to assign? to C_3.

What makes a good cluster?

In supervised learning, we had a very good idea what made a good prediction function: Loss functions, misclassification rates, actual costs, etc.

What makes a good clustering?

- Tightly packed groups? → within group small distances
- Well-separated groups? → between group large dist.

You'll eventually find that a clustering is "good" if it turns out to be *useful*, usually for some downstream purpose.

We'll explore several versions of all of these notions, each of which will be useful...sometimes.

Within-cluster scatter

One measure of a clustering is within-cluster scatter. This is a measure of how spread out the points are within each cluster.

The general notion is that a good clustering should lead to tightly-packed clusters with low within-cluster scatter.

Notation: $C(i) = k$ means that X_i is assigned to group k, and n_k is the number of points in the group k. Also, let $\underline{\underline{d_{ij}}} = d(X_i, X_j)$.

The within-cluster scatter is defined as

for a fixed k,

$$W = \frac{1}{2} \sum_{k=1}^{K} \underbrace{\frac{1}{n_k} \sum_{C(i)=k,\, C(j)=k} d_{ij}}_{}$$

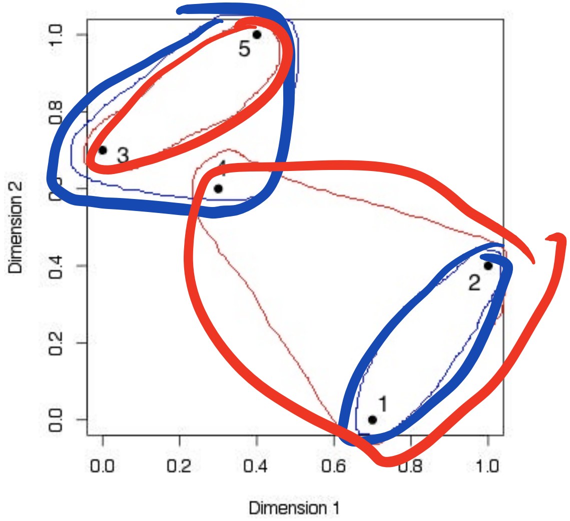
Smaller W is better

Simple example

Here $n = 5$ and $K = 2$,
$X_i \in \mathbb{R}^2$ and $d_{ij} = \|X_i - X_j\|_2^2$

pairwise dist² between pts.

	1	2	3	4	5
1	0	0.25	0.98	0.52	1.09
2	0.25	0	1.09	0.53	0.72
3	0.98	1.09	0	0.10	0.25
4	0.52	0.53	0.10	0	0.17
5	1.09	0.72	0.25	0.17	0



- Red clustering:
  $$\underline{W_{\text{red}}} = (\overset{d_{12}}{0.25} + \overset{d_{24}}{0.53} + \underset{d_{14}}{0.52})/3 + \overset{d_{35}}{0.25}/2 = 0.56$$
- Blue clustering:
  $$W_{\text{blue}} = 0.25/2 + (0.10 + 0.17 + 0.25)/3 = 0.30$$

} smaller within cluster scatter.

Finding the best group assignments

Smaller W is better, so why don't we just directly find the clustering C that minimizes W?

Problem: doing so requires trying all possible assignments of the n points into K groups. The number of possible assignments is huge!

For 25 points and 4 groups: $\approx \boxed{5 \times 10^{13}}$

Most problems we look at are going to have way more than $n = 25$ observations, and potentially more than $K = 4$ clusters too.

So we'll have to settle for an approximation

Fact:

$$\frac{1}{2}\left[\frac{1}{n_k}\sum_{C(i)=k,\,C(j)=k} d_{ij}\right] = \underbrace{\sum_{C(i)=k} d\left(x_i, \frac{1}{n_k}\sum_{C(j)=k} x_j\right)}_{\text{variance of cluster.}}$$

Rewriting the within-cluster scatter

Focus on Euclidean space: now $X_i \in \mathbb{R}^p$ and dissimilarities are $d(X_i, X_j) = \|X_i - X_j\|_2^2$

Fact: within-cluster scatter can be rewritten as

$$\underbrace{\frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{C(i)=k} \sum_{C(j)=k} \|X_i - X_j\|_2^2}_{\text{wc scatter}} = \underbrace{\sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - \bar{X}_k\|_2^2}_{\text{wc variance}}$$

with $\bar{X}_k$ the average of points in group k, $\bar{X}_k = \frac{1}{n_k} \sum_{C(i)=k} X_i$. The right-hand side above is called within-cluster variation

Hence, equivalently we seek a clustering C that minimizes the within-cluster variation (approximately)

Rewriting the minimization

centroids

Remember: we want to choose C to minimize

$$\min_{C} W(C) = \min_{C} \sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - \bar{X}_k\|_2^2$$

$$\min_{\text{clustering}} \min_{c_1,\ldots,c_K} \sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - c_k\|_2^2$$

Question: for any $Z_1, \ldots Z_m \in \mathbb{R}^p$, suppose that we minimize the quantity $\sum_{i=1}^m \|Z_i - c\|_2^2$ over c. What is the minimizing c?

$$c = \frac{1}{m} \sum_{i=1}^{m} Z_i .$$

Rewriting the minimization

With the fact from the last slide, we can introduce new variables c_k, and note that minimizing

$$\sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - \bar{X}_k\|_2^2$$

is the same as minimizing

$$\underbrace{\sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - c_k\|_2^2,}$$

over both clusterings C and $c_1, \ldots c_K \in \mathbb{R}^p$

Finally...

We want to minimize

$$\sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - c_k\|_2^2,$$

→ alternating min.

over both clusterings C and $c_1, \ldots c_K \in \mathbb{R}^p$. It's still not clear how to do this. However, can you:

Minimize it just over C? → assign i to nearest $\{c_1, \ldots, c_K\}$.

Minimize it just over c_k? → c_k assigned mean of all pts $C(i) = k$.

K-means algorithm

The K-means clustering algorithm approximately minimizes the enlarged criterion by alternately minimizing over C and $c_1, \ldots c_K$

We start with an initial guess for $c_1, \ldots c_K$ (e.g., pick K points at random over the range of $X_1, \ldots X_n$), then repeat:

1. Minimize over C: for each $i = 1, \ldots n$, find the cluster center c_k closest to X_i, and let $C(i) = k$
2. Minimize over $c_1, \ldots c_K$: for each $k = 1, \ldots K$, let $c_k = \bar{X}_k$, the average of points in group k

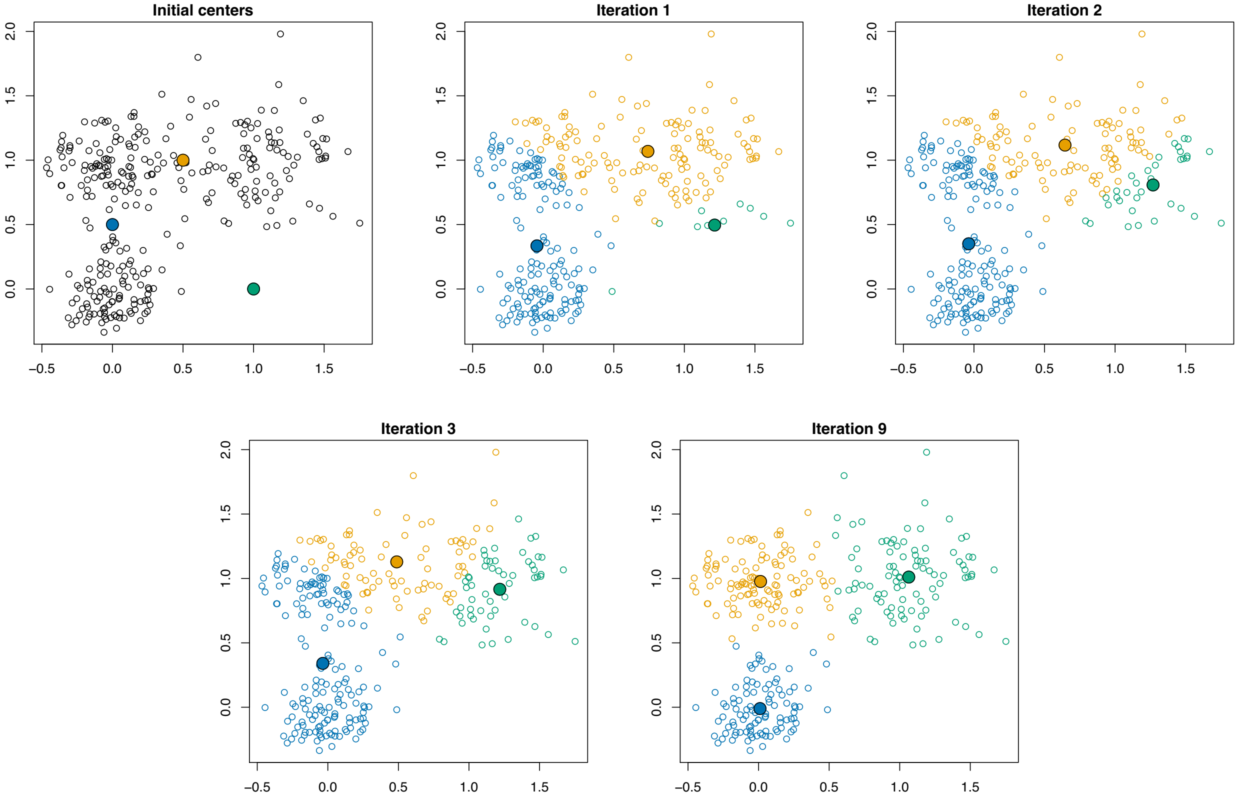
Stop when within-cluster variation doesn't change

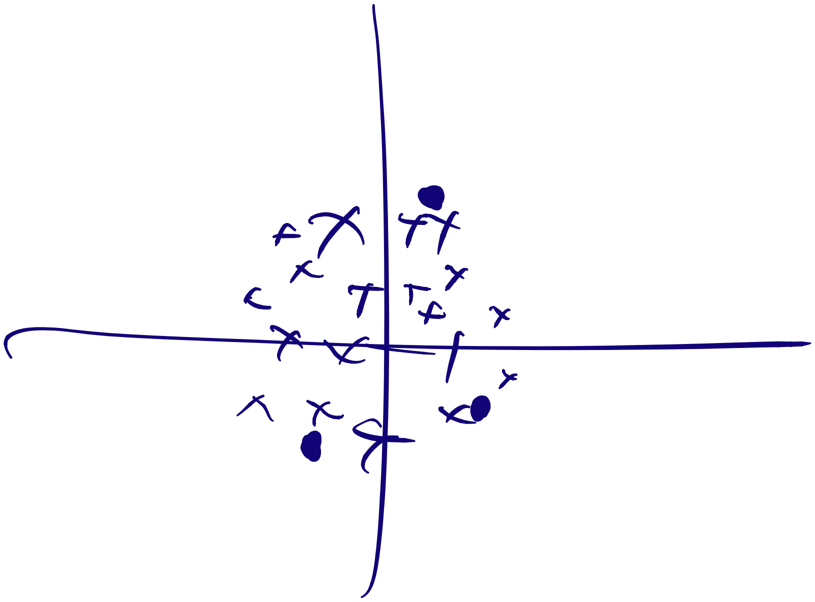
In words:

1. Cluster (label) each point based the closest center
2. Replace each center by the average of points in its cluster

K-means example

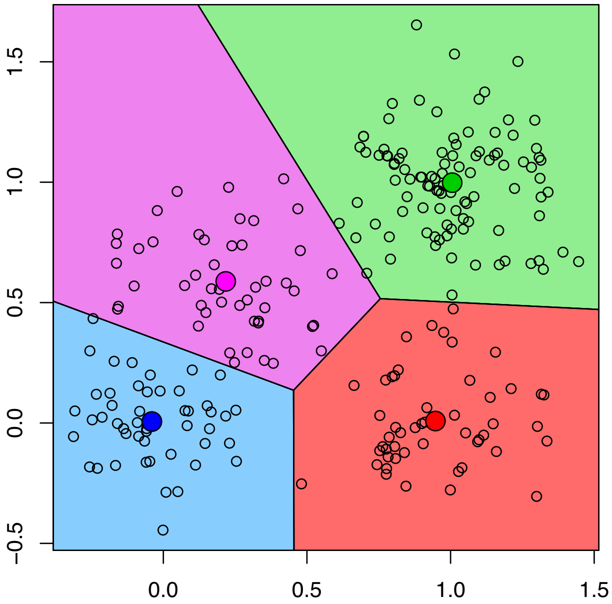
Here $X_i \in \mathbb{R}^2$, $n = 300$, and $K = 3$





Voronoi tessellation

Given cluster centers, we identify each point to its nearest center. This defines a Voronoi tessellation of $\mathbb{R}^p$



Given $c_1, \ldots c_K \in \mathbb{R}^p$, we define the Voronoi sets

$$V_k = \{x \in \mathbb{R}^p : \|x - c_k\|_2^2 \leq \|x - c_j\|_2^2,\ j = 1, \ldots K\},\ \ k = 1, \ldots K$$

These are convex polyhedra (should remind you of LDA)

Properties of K-means

- Within-cluster variation decreases with each iteration of the algorithm. I.e., if W_t is the within-cluster variation at iteration t, then $\boxed{W_{t+1} \leq W_t}$
- The algorithm always converges, no matter the initial cluster centers. In fact, it takes $\leq K^n$ iterations (why?)

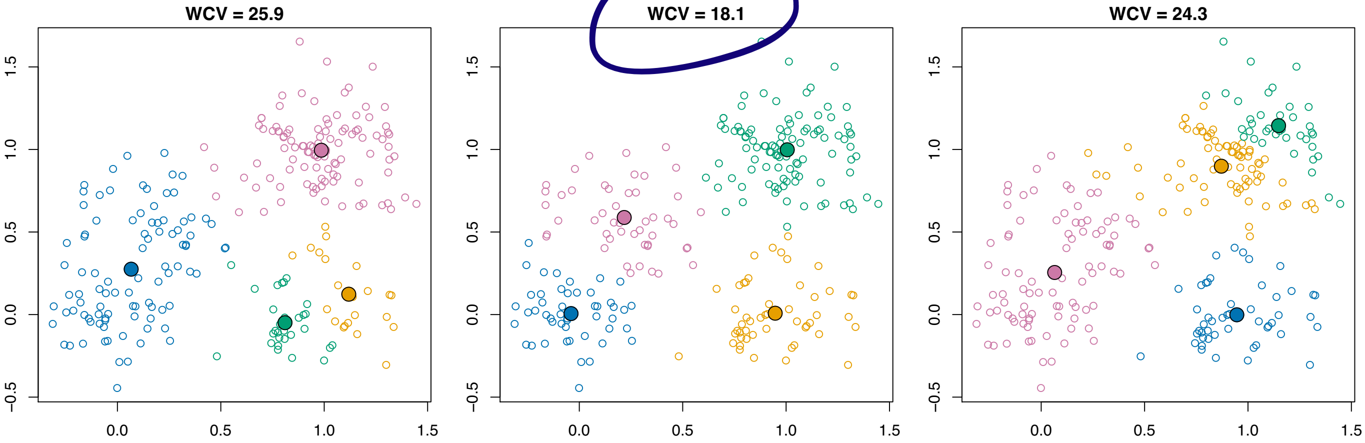
k-means++
↓
clever way to initialize

- The final clustering depends on the initial cluster centers. Sometimes, different initial centers lead to very different final outputs. So we typically run K-means multiple times (e.g., 10 times), randomly initializing cluster centers for each run, then choose among from collection of centers based on which one gives the smallest within-cluster variation
- The algorithm is not guaranteed to deliver the clustering that globally minimizes within-cluster variation (recall: this would require looking through all possible assignments!)

K-means example, multiple runs

Here $X_i \in \mathbb{R}^2$, $n = 250$, and $K = 4$, the points are not as well-separated

pick this clustering.



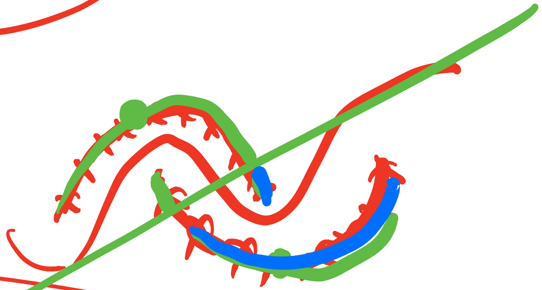
These are results of result of running the K-means algorithm with different initial centers (chosen randomly over the range of the X_i's). We choose the second collection of centers because it yields the smallest within-cluster variation

What are some things K-means lacks?

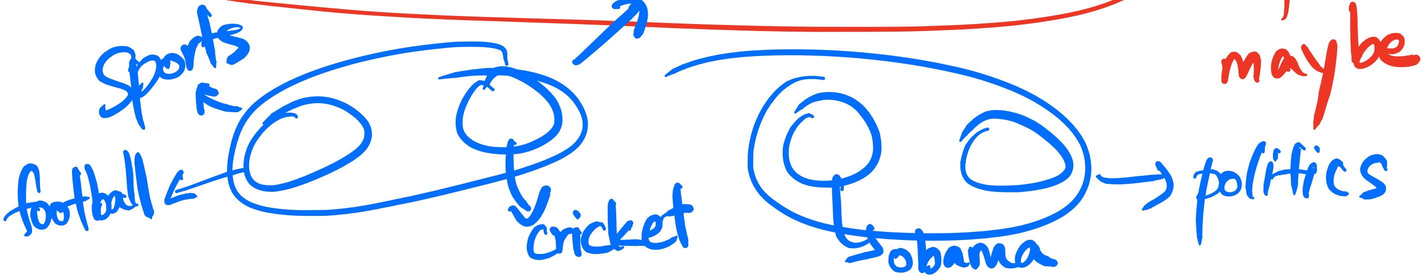
K-means is a famous, standard clustering algorithm. However, it lacks several potentially-desirable qualities:

- Ability to use other measures of dissimilarity
- "Interpretable" cluster centers
- Deterministic results
- Multi-level/scale view of clusters, Nested clusters.

want to use diff dist.



maybe



In K-means, cluster centers are averages

A cluster center is representative for all points in a cluster, also called a prototype

In K-means, we simply take a cluster center to be the average of points in the cluster. Great for computational purposes—but how does it lend to interpretation?

Sometimes we prefer methods that return a representative item for the cluster, rather than an average. For example: a "typical" asset or company that is similar to all the other members of the cluster. This makes it easier to think about what the cluster means.

Suppose we were clustering documents. What does an "average" document mean? A typical document is more useful.

K-medoids algorithm

K-medoiids clustering addresses the first two concerns. It make each center one of the cluster points. It also allows other dimilarities to be substituted.

Initial guess for centers $c_1, \ldots c_K$ (e.g., randomly select K of the points $X_1, \ldots X_n$), then repeat:

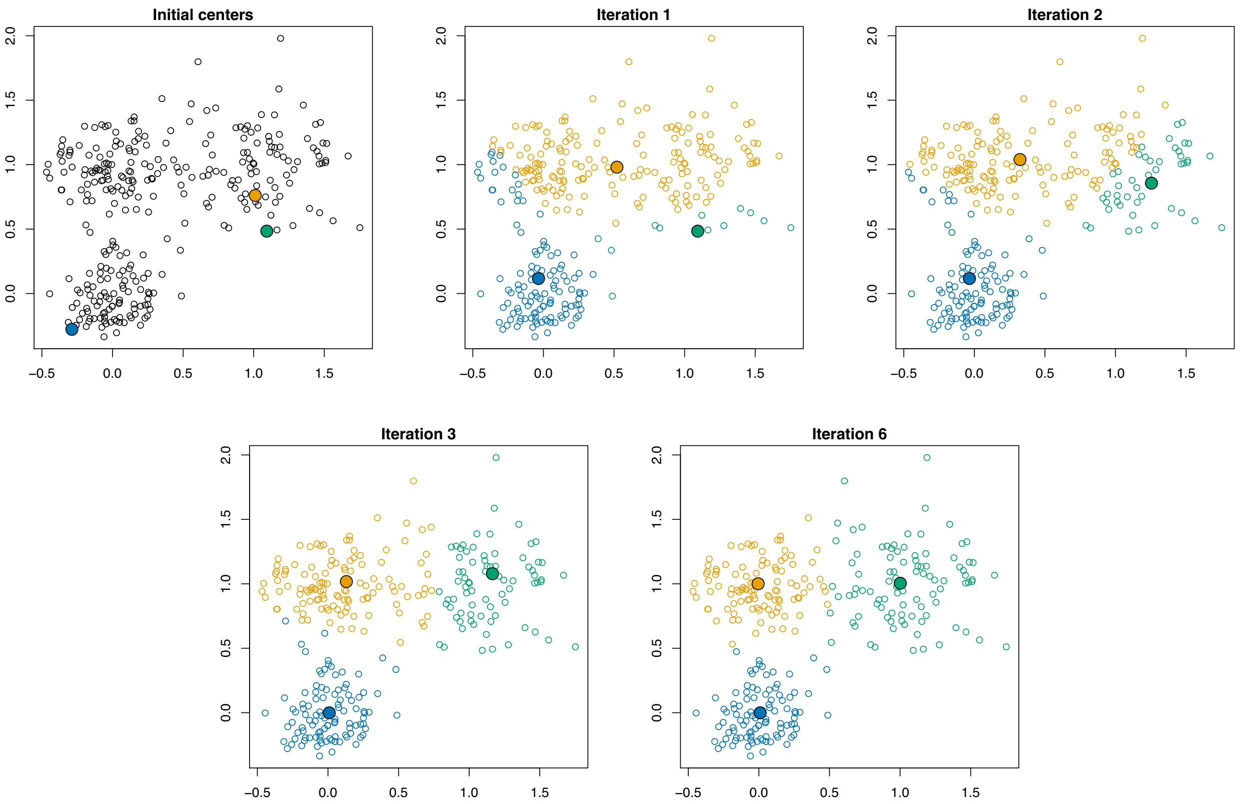
1. Minimize over C: for each $i = 1, \ldots n$, find the cluster center c_k closest to X_i, and let $C(i) = k$
2. Minimize over $c_1, \ldots c_K$: for each $k = 1, \ldots K$, let $c_k = X_k^*$, the medoid of points in cluster k, i.e., the point X_i in cluster k that minimizes $\sum_{C(j)=k} \|X_j - X_i\|_2^2$

Stop when within-cluster variation doesn't change

In words:

1. Cluster (label) each point based on the closest center
2. Replace each center by the medoid of points in its cluster

K-medoids example



Note: only 3 points had different labels under K-means

Properties of K-medoids

The K-medoids algorithm shares the properties of K-means that we discussed (each iteration decreases the criterion; the algorithm always converges; different starts gives different final answers; it does not achieve the global minimum)

K-medoids returns centers that are actual data points.

K-medoids generally returns a higher value of $\sum_{k=1}^{K} \sum_{C(i)=k} \|X_i - c_k\|_2^2$ than does K-means (why?).

K-medoids is computationally harder than K-means (because of step 2: computing the medoid is harder than computing the average)